

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平5-81433

(43)公開日 平成5年(1993)4月2日

(51)Int.Cl.
G 0 6 F 15/70
15/332
15/68

識別記号 庁内整理番号
4 5 5 A 9071-5L
B 6798-5L
S 6798-5L
K 8420-5L

F I

技術表示箇所

審査請求 未請求 請求項の数3(全11頁)

(21)出願番号 特願平3-268830

(22)出願日 平成3年(1991)9月20日

(71)出願人 000010076

ヤマハ発動機株式会社
静岡県磐田市新貝2500番地

(72)発明者 山口 昌志

静岡県磐田市新貝2500番地ヤマハ発動機株
式会社内

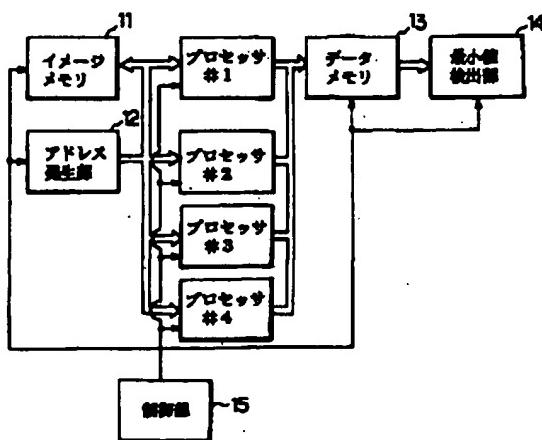
(74)代理人 弁理士 山下 充一

(54)【発明の名称】 パターン・マッチング法及び装置

(57)【要約】

【目的】 处理時間を短縮して高速化を実現することができるパターン・マッチング法及び装置を提供すること。

【構成】 处理画像を複数の処理領域に分割し、各処理領域毎に別々のプロセッサで画像処理する並列処理法と、各処理領域での画像処理において対象画像のテンプレート画像に対するミス・マッチ度が所定の閾値を超えると処理を打切るSSDA法を併用するとともに、前記並列処理法における各プロセッサの処理領域に位置的な片寄りがないように処理画像を分割する。従って、各プロセッサでのSSDA法による処理時間にバラツキが生じず、処理時間が短縮されて処理の高速化が図られる。又、イメージメモリ11と、アドレス発生部12と、複数のプロセッサ#1～#4と、データメモリ13と、最小値検出部14と、制御部15を含んでパターン・マッチング装置を構成する。



【特許請求の範囲】

【請求項1】 読取り対象画像がテンプレート画像と一致するマッチング領域を求めるパターン・マッチング法において、処理画像を複数の処理領域に分割し、各処理領域毎に別々のプロセッサで画像処理する並列処理法と、各処理領域での画像処理において対象画像のテンプレート画像に対するミス・マッチ度が所定の閾値を超えると処理を打切るSSDA法を併用するとともに、前記並列処理法における各プロセッサの処理領域に位置的な片寄りがないように処理画像を分割するようにしたことを特徴とするパターン・マッチング法。

【請求項2】 読取り対象画像とテンプレート画像にアダマール変換を施し、両画像のアダマール係数を用いて両画像のミス・マッチ度を求める方法を採用し、前記並列処理法において各処理領域の1ブロック目ののみの処理によってそのブロックのミス・マッチ度を求め、このミス・マッチ度が所定の閾値以下となる候補点を求め、求められた候補点を各プロセッサに対して振り分けることを特徴とする請求項1記載のパターン・マッチング法。

【請求項3】 読取られた対象画像を保管するイメージメモリと、処理画像上にアドレスを発生させるアドレス発生部と、複数に分割された処理領域の各々を独立に画像処理する複数のプロセッサと、各プロセッサによる画像処理によって得られたミス・マッチ度とアドレスのデータを保管するデータメモリと、ミス・マッチ度の最小値を検出する最小値検出部と、これらイメージメモリ、アドレス発生部、プロセッサ、データメモリ及び最小値検出部の動作を制御する制御部を含んで構成されることを特徴とするパターン・マッチング装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、読み取り対象画像がテンプレート画像と一致するマッチング領域を求めるパターン・マッチング法及び装置に関する。

【0002】

【従来の技術】 画像処理における高速化の手法の1つとして、並列処理法が知られている。この並列処理法は、図12に示すような処理画像を複数（図示例では、4つ）の処理領域に分割し、各処理領域毎に別々のプロセッサで処理することによって高速化を図る方法であって、プロセッサ数（処理領域数）をnとすると、処理時間は $1/n$ 倍に短縮される。

【0003】一方、パターン・マッチング法においては、処理の高速化を図る手法としてSSDA法と呼ばれる方法が知られている。この方法は、図13に示すように、対象画像のテンプレート画像に対するミス・マッチ度の計算過程でミス・マッチ度が予め設定された閾値を超えた場合には、その場所は最早マッチングポイントではないものと判断し、そこで処理を打切ることによって高速化を図る方法である。

【0004】

【発明が解決しようとする課題】 ところで、パターン・マッチング法において高速化を実現するために前記並列処理法とSSDA法を組み合わせて用いることが考えられる。

【0005】しかしながら、図13から明らかなように、処理領域によって処理時間が異なるため、例えば図12に示すように処理画像を4分割した場合には、並列処理による各プロセッサ#1、#2、#3、#4の処理時間が図14に示すように異なり、全体の処理時間は最も遅いプロセッサ#2の処理時間となってしまう。このため、他のプロセッサ#1、#3、#4はロストタイムを持つこととなり、並列処理法のメリットを十分活かすことができない。

【0006】本発明は上記問題に鑑みてなされたもので、その目的とする処は、処理時間を短縮して高速化を実現することができるパターン・マッチング法及び装置を提供することにある。

【0007】

【課題を解決するための手段】 上記目的を達成すべく本発明は、読み取り対象画像がテンプレート画像と一致するマッチング領域を求めるパターン・マッチング法において、処理画像を複数の処理領域に分割し、各処理領域毎に別々のプロセッサで画像処理する並列処理法と、各処理領域での画像処理において対象画像のテンプレート画像に対するミス・マッチ度が所定の閾値を超えると処理を打切るSSDA法を併用するとともに、前記並列処理法における各プロセッサの処理領域に位置的な片寄りがないように処理画像を分割するようにしたことを特徴とする。そして、本方法において、読み取り対象画像とテンプレート画像にアダマール変換を施し、両画像のアダマール係数を用いて両画像のミス・マッチ度を求める方法を採用し、前記並列処理法において各処理領域の1ブロック目ののみの処理によってそのブロックのミス・マッチ度を求め、このミス・マッチ度が所定の閾値以下となる候補点を求め、求められた候補点を各プロセッサに対して振り分けることも特徴とする。

【0008】又、本発明は、読み取られた対象画像を保管するイメージメモリと、処理画像上にアドレスを発生させるアドレス発生部と、複数に分割された処理領域の各々を独立に画像処理する複数のプロセッサと、各プロセッサによる画像処理によって得られたミス・マッチ度とアドレスのデータを保管するデータメモリと、ミス・マッチ度の最小値を検出する最小値検出部と、これらイメージメモリ、アドレス発生部、プロセッサ、データメモリ及び最小値検出部の動作を制御する制御部を含んでパターン・マッチング装置を構成したことを特徴とする。

【0009】

【作用】 本発明によれば、パターン・マッチング法に並列処理法とSSDA法が併用され、各プロセッサでの処

理においては、各プロセッサが処理すべき領域が処理画像全体に対して位置的に片寄らないよう振り分けられるため、各プロセッサでのSSDA法による処理時間にバラツキが生じず、ロスタイムがなくなって効率的な並列処理がなされ、この結果、処理時間が短縮されて高速化が実現される。

【0010】

【実施例】以下に本発明の第1実施例を添付図面に基づいて説明する。

【0011】図1は第1実施例に係るパターン・マッチング装置の構成を示すブロック図である。該パターン・マッチング装置は、読み取られた対象画像を保管するイメージメモリ11と、処理画像上にアドレスを発生させるアドレス発生部12と、複数に分割された処理領域の各々を独立に画像処理する複数のプロセッサ#1、#2、#3、#4と、各プロセッサ#1～#4による画像処理によって得られたミス・マッチ度とアドレスのデータを保管するデータメモリ13と、ミス・マッチ度の最小値を検出する最小値検出部14と、これらイメージメモリ11、アドレス発生部12、プロセッサ#1～#4、データメモリ13及び最小値検出部14の動作を制御する制御部15を含んで構成されている。

【0012】以下に上記パターン・マッチング装置を用いて実施される本発明に係るパターン・マッチング法を図2乃至図4に基づいて説明する。尚、図2は前記制御部15による処理手順を示すフローチャート、図3は各プロセッサ#1～#4での処理手順を示すフローチャート、図4(a)、(b)、(c)、(d)は処理画像の分割の様子を示す図である。

【0013】本実施例に係るパターン・マッチング法においては、処理画像が図4(a)に示すように乱数1、2、3、4によって分割され、図中、乱数1、2、3、4で示される領域は各々プロセッサ#1、#2、#3、#4によって処理される。

【0014】即ち、先ず処理画像について乱数1～4によるアドレスが発生したか否かがチェックされ(図2のステップ1)、アドレスが処理画像全体について発生していないなければ、プロセッサ#1～#4のうちで処理をしていないものがあるか否かがチェックされ(図2のステップ2)、処理をしていないプロセッサがあれば、乱数1～4によるアドレスを発生させ(図2のステップ3)、発生したアドレスにダブリがないことを確認した後(図2のステップ4)、処理をしていないプロセッサについて対応する領域について処理を開始させる(図2のステップ5)。

【0015】そして、処理画像全体についてアドレスが発生し、且つ全てのプロセッサ#1～#4が処理を開始すると、全てのプロセッサ#1～#4での処理が終了したか否かがチェックされる(図2のステップ6)。ここで、各プロセッサ#1～#4での画像処理を図3に従っ

て説明する。

【0016】即ち、従前に求められていたミス・マッチ度がクリアされ(図3のステップ1)、テンプレート画像全体について処理が終了したか否かがチェックされ(図3のステップ2)、終了していないければ、処理する画素を更新してミス・マッチ度の差分値の絶対値が求められ(図3のステップ3)、その差分値がミス・マッチ度に加算されて新たにミス・マッチ度が算出される(図3のステップ4)。そして、SSDA法によって、この算出されたミス・マッチ度が予め設定された閾値より大きいか否かがチェックされ(図3のステップ5)、ミス・マッチ度が閾値を超れば、その時点でその場所は最早マッチングポイントではないものと判断し、処理は直ちに打ち切られる(図3のステップ7)。これに対し、ミス・マッチ度が閾値より小さい間はテンプレート画像全体について処理が終了するまで以上の一連の処理(図3のステップ2～5の処理)が繰り返される。

【0017】そして、テンプレート画像全体について処理が終了すると、求められたミス・マッチ度とアドレスが図1に示すデータメモリ13に書き込まれ(図3のステップ8)、各プロセッサ#1～#4での処理が終了する(図3のステップ7)。

【0018】以上の各プロセッサ#1～#4での処理においては、各プロセッサ#1～#4が処理すべき領域は図4(a)に示すように乱数1～4によって割り当てられ、これらは処理画像全体に対して位置的な片寄りを生じないため、各プロセッサ#1～#4でのSSDA法による処理時間にバラツキが生じず、ロスタイムがなくなって効率的な並列処理がなされる。

【0019】而して、全てのプロセッサ#1～#4での上記処理が終了すると、図1に示す最小値検出部14によってミス・マッチ度の最小値が検出され、その最小値を示す位置がマッチングポイントとされて(図2のステップ7)一連のパターン・マッチングの処理が終了する(図2のステップ8)。

【0020】尚、各プロセッサ#1～#4の処理領域が処理画像全体に対して位置的な片寄りを生じないような分割法としては、図4(b)に示すように縦に細かく分割する方法、図4(c)に示すように横に細かく分割する方法、図4(d)に示すように斜めに細かく分割する方法等がある(図中、数字1、2、3、4はそれぞれプロセッサ#1、#2、#3、#4での処理領域を示す)。

【0021】次に、本発明の第2実施例を添付図面に基づいて説明する。

【0022】図5は第2実施例に係るパターン・マッチング装置の構成を示すブロック図であり、該装置は図1に示した前記第1実施例に係る装置に候補リスト記憶部16を付加したものであって、他の構成は第1実施例に係る装置のそれと同様である。

【0023】以下に上記パターン・マッチング装置を用いて実施されるパターン・マッチング法を図8乃至図11に基づいて説明する。尚、図6は1次マッチングの処理手順を示すフローチャート、図7は処理画像上のクラスタの分布を示す図、図8は図7のA部（クラスタ1）における候補点の振り分けを示す図、図9は簡便な候補点の振り分け方法を示す図、図10及び図11は2次マッチングの処理手順を示すフローチャートである。

【0024】而して、本実施例に係るパターン・マッチング法は、読み取り対象画像とテンプレート画像にアダマール変換を施し、両画像のアダマール係数を用いて両画像のミス・マッチ度を求める方法（詳細は特開平2-83786号公報参照）を採用している。

【0025】ところで、アダマール変換を用いた場合、位置(m, n)におけるアダマール係数はこの位置(m, n)に連続する位置(m-1, n)での値を用いると計算量が $6/15$ に削減される。つまり、或るアドレスブロックのアダマール係数は、アドレスが連続するブロックのアダマール係数を利用して高速に求めることができる。

【0026】ところが、SSDA法を適用すると、2ブロック目以降はアドレスが不連続となる可能性があり、前述のメリットを活かすことができない。そこで、本実施例では、処理を次の1次マッチングと2次マッチングの2ステップに分けて効率的な並列処理を行なうようにしている。

(1) 1次マッチング

本手法では、前記第1実施例と同様に処理画像が複数(4つ)の処理領域に分割され、各処理領域毎に異なるプロセッサ#1～#4でアダマール変換を用いた並列処理がなされる。但し、ここではテンプレート画像全体に対する処理は行なわれず、1ブロック目のみの処理が行なわれる（図6のステップ1～3）。

【0027】そして、1ブロック目の処理によって算出されたミス・マッチ度が予め設定された閾値と比較され（図6のステップ4）、ミス・マッチ度が閾値より小さければ、そのミス・マッチ度とアドレスが候補点として候補リスト記憶部16に書き込まれ（図6のステップ5）、処理領域全体について以上の処理がなされると、1次マッチングの処理が終了する（図6のステップ6）。

【0028】以上のように、1次マッチングにおいては最初の1ブロック目のみの処理がなされるだけであり、SSDA法による処理の打切りがなく、しかも処理アドレスが連続しているため、各プロセッサ#1～#4での処理時間が等しくなり、且つ短縮される。

(2) 2次マッチング

前記1次マッチングによって図7に示す候補点群（以下、クラスタ1, 2, 3と称す）が求められるが、これらのクラスタ1, 2, 3は処理画像全体に一様に分布す

るのでなく、幾つか散在するように分布する。そして、各クラスタ1～3は1つのパターン（図7に示す破線にて囲まれる領域）に対応した候補点となるため、SSDA法による打切りブロック数は各クラスタ1～3内では略同程度となる。従って、各プロセッサ#1～#4に対する候補点の振り分けを図8に示すように規則的に行なえば、各プロセッサ#1～#4での処理時間が略等しくなり、効率的な並列処理を行なうことができる。

尚、図8において、①, ②, ③, ④はそれぞれプロセッサ#1～#4によって処理されるべき候補点を示す。

【0029】ところで、図8に示すように、処理領域の或る部分のみにおいて候補点をクラスタリングするには特別な処理が必要である。そこで、図9に示すように、処理領域の左上から①, ②, ③, ④の順に規則的に振り分けるようにしても良く、この場合1つのクラスタに注目すると、このクラスタ内には各プロセッサ#1～#4に振り分けられるべき候補点の数（①, ②, ③, ④の数）が略均等に分布するため、効率的な並列処理が可能となる。

【0030】ここで、2次マッチングの処理手順を図10及び図11に従って説明する。

【0031】即ち、候補リスト全てについての処理が終了したか否かがチェックされ（図10のステップ1）、終了していないければ、処理をしていないプロセッサがあるか否かチェックされ（図10のステップ2）、処理をしていないプロセッサがあれば、図5に示す候補リスト記憶部16から候補リストが読み込まれて処理アドレスが決定され（図10のステップ3）、その処理をしていないプロセッサに2次マッチング処理を開始させる（図10のステップ4）。この処理は候補リストの全てについて繰り返され、候補リストの全てについて処理が終了すると、全てのプロセッサ#1～#4での処理が終了したか否かがチェックされる（図10のステップ5）。

【0032】ここで、各プロセッサ#1～#4での2次マッチングの処理手順を図11に基いて説明する。

【0033】即ち、各プロセッサ#1～#4では前記1次マッチングによって求められた1ブロック目のミス・マッチ度が初期のミス・マッチ度とされ（図11のステップ1）、処理するブロックを更新しながらそのブロックのミス・マッチ度が求められ（図11のステップ3）、初期のミス・マッチ度にそのブロックのミス・マッチ度が加算されて新しいミス・マッチ度が算出される（図11のステップ4）。そして、SSDA法によって、この算出されたミス・マッチ度が予め設定された閾値より大きいか否かがチェックされ（図11のステップ5）、ミス・マッチ度が閾値を超えるか、その時点でその場所は最早マッチングポイントではないものと判断し、処理は直ちに打ち切られる（図11のステップ7）。

これに対し、ミス・マッチ度が閾値より小さい間は全てのブロックについて処理が終了するまで以上の一連の処

理(図11のステップ2~5の処理)が繰り返される。

【0034】そして、全てのプロックについて処理が終了すると、求められたミス・マッチ度とアドレスが図5に示すデータメモリ13に書き込まれ(図11のステップ6)、各プロセッサ#1~#4での処理が終了する(図11のステップ7)。

【0035】而して、全てのプロセッサ#1~#4での上記処理が終了すると、図5に示す最小値検出部14によってミス・マッチ度の最小値が検出され、その最小値を示す位置がマッチングポイントとされて(図10のステップ6)2次マッチングの処理が終了する(図10のステップ7)。

【0036】

【発明の効果】以上の説明で明らかな如く、本発明によれば、読み取り対象画像がテンプレート画像と一致するマッチング領域を求めるパターン・マッチング法において、処理画像を複数の処理領域に分割し、各処理領域毎に別々のプロセッサで画像処理する並列処理法と、各処理領域での画像処理において対象画像のテンプレート画像に対するミス・マッチ度が所定の閾値を超えると処理を打切るSSDA法を併用するとともに、前記並列処理法における各プロセッサの処理領域に位置的な片寄りがないように処理画像を分割するようにしたため、パターン・マッチング処理における処理時間を短縮して処理の高速化を実現することができるという効果が得られる。

【図面の簡単な説明】

【図1】本発明の第1実施例に係るパターン・マッチング装置の構成を示すブロック図である。

【図2】本発明の第1実施例に係るパターン・マッチング装置の制御部による処理手順を示すフローチャートで*30

*ある。

【図3】本発明の第1実施例に係るパターン・マッチング装置の各プロセッサでの処理手順を示すフローチャートである。

【図4】(a), (b), (c), (d)は処理画像の分割の様子を示す図である。

【図5】本発明の第2実施例に係るパターンマッチング装置の構成を示すブロック図である。

【図6】1次マッチングの処理手順を示すフローチャートである。

【図7】処理画像上のクラストの分布を示す図である。

【図8】図7のA部(クラスト1)における候補点の振り分けを示す図である。

【図9】簡便な候補点の振り分け方法を示す図である。

【図10】2次マッチングの処理手順を示すフローチャートである。

【図11】2次マッチングの処理手順を示すフローチャートである。

【図12】並列処理法における処理画像の分割を示す図である。

【図13】SSDA法における処理回数とミス・マッチ度との関係を示す図である。

【図14】各プロセッサの処理時間を示す図である。

【符号の説明】

11 イメージメモリ

12 アドレス発生部

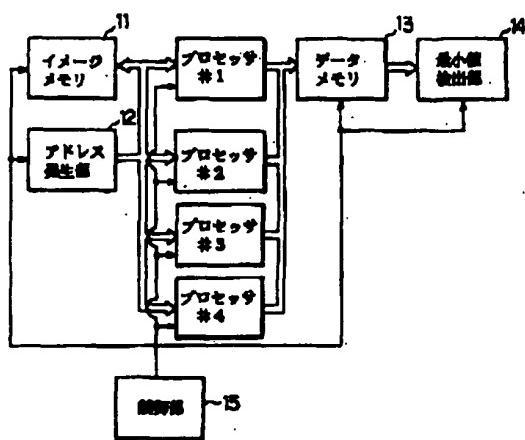
13 データメモリ

14 最小値検出部

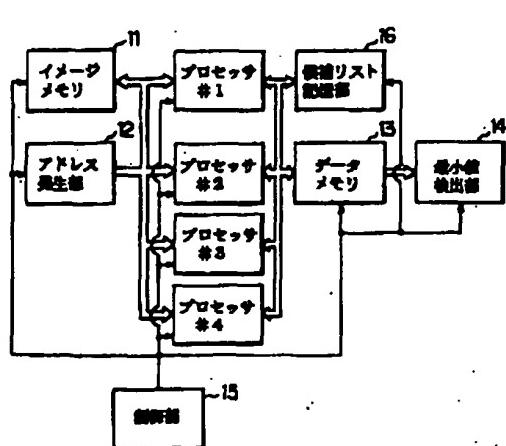
15 制御部

16 候補リスト記憶部

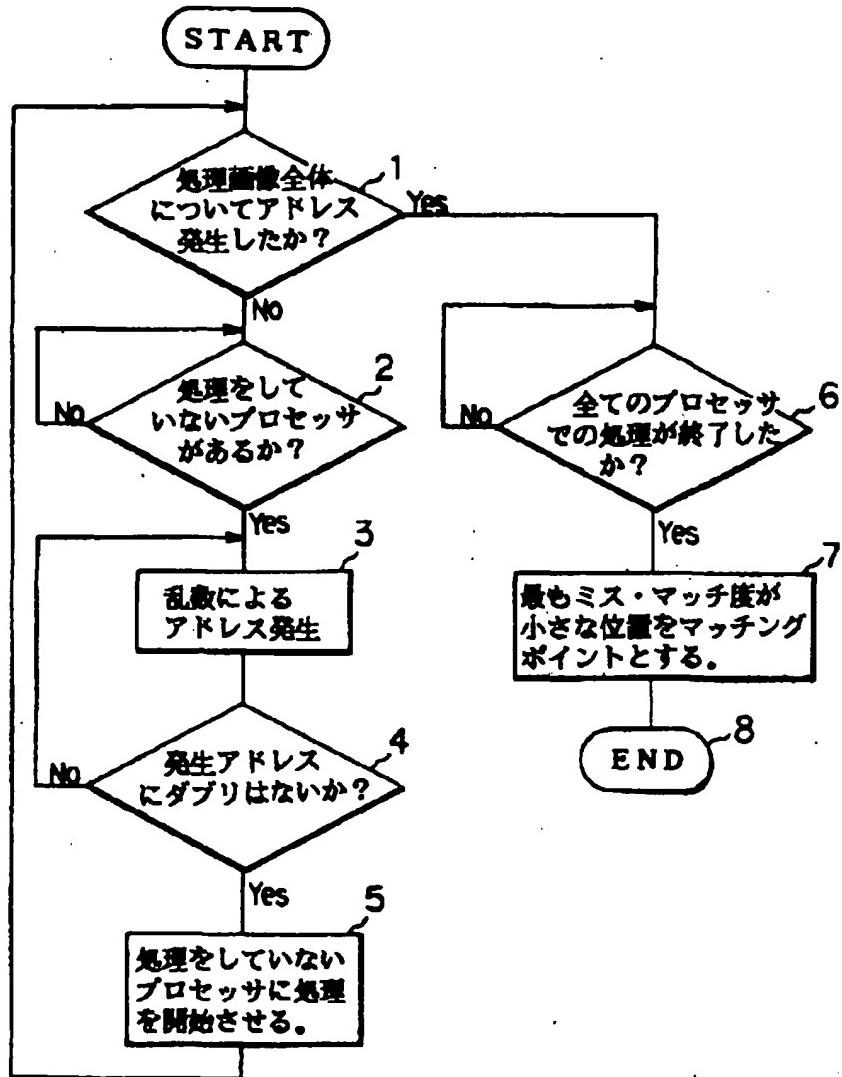
【図1】



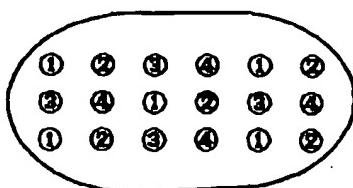
【図5】



【図2】

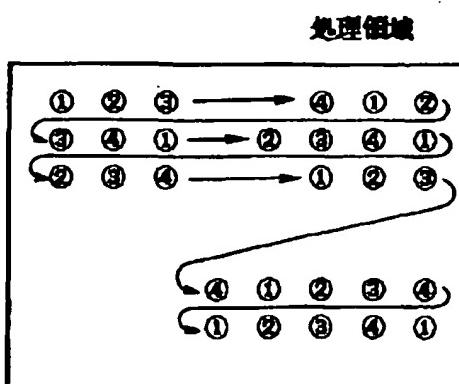


【図8】

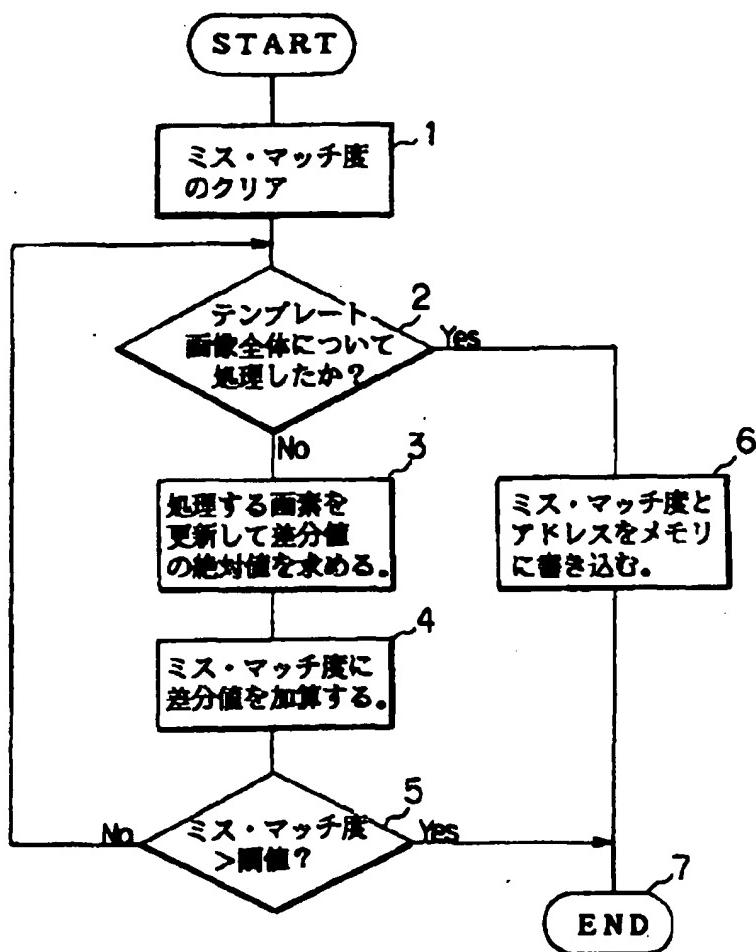


- ①: プロセッサ#1で処理
- ②: プロセッサ#2で処理
- ③: プロセッサ#3で処理
- ④: プロセッサ#4で処理

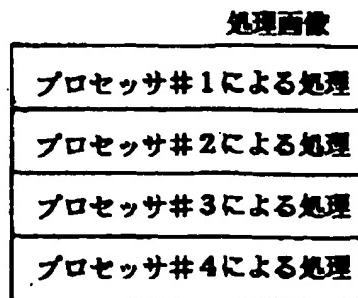
【図9】



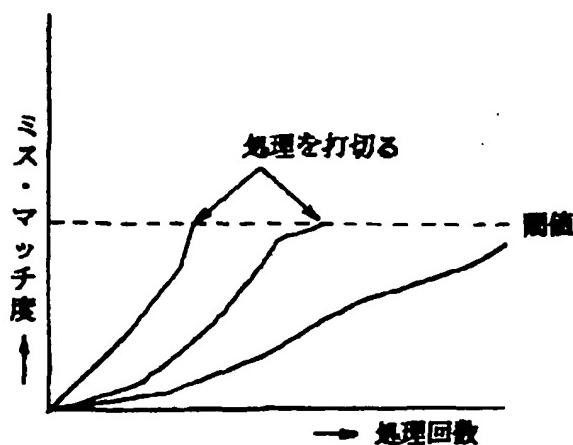
【図3】



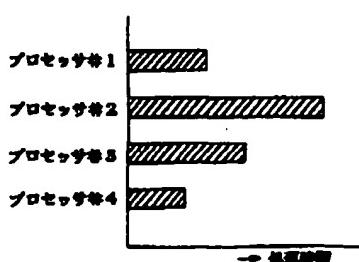
【図12】



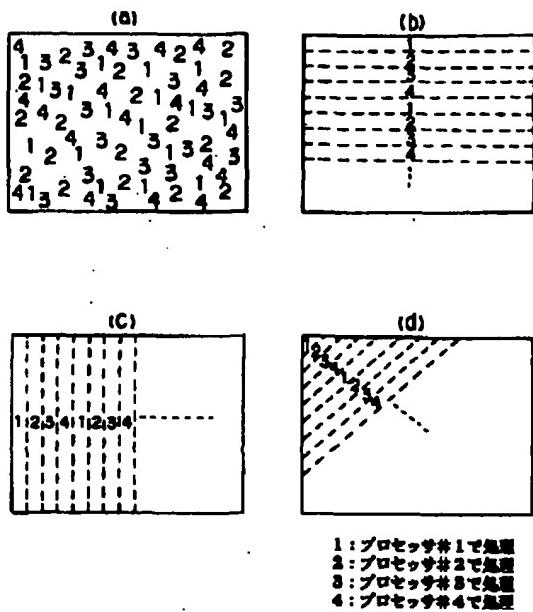
【図13】



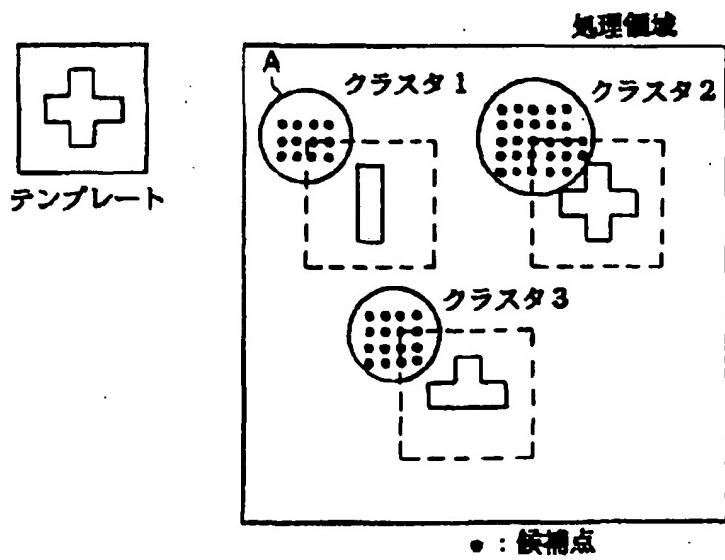
【図14】



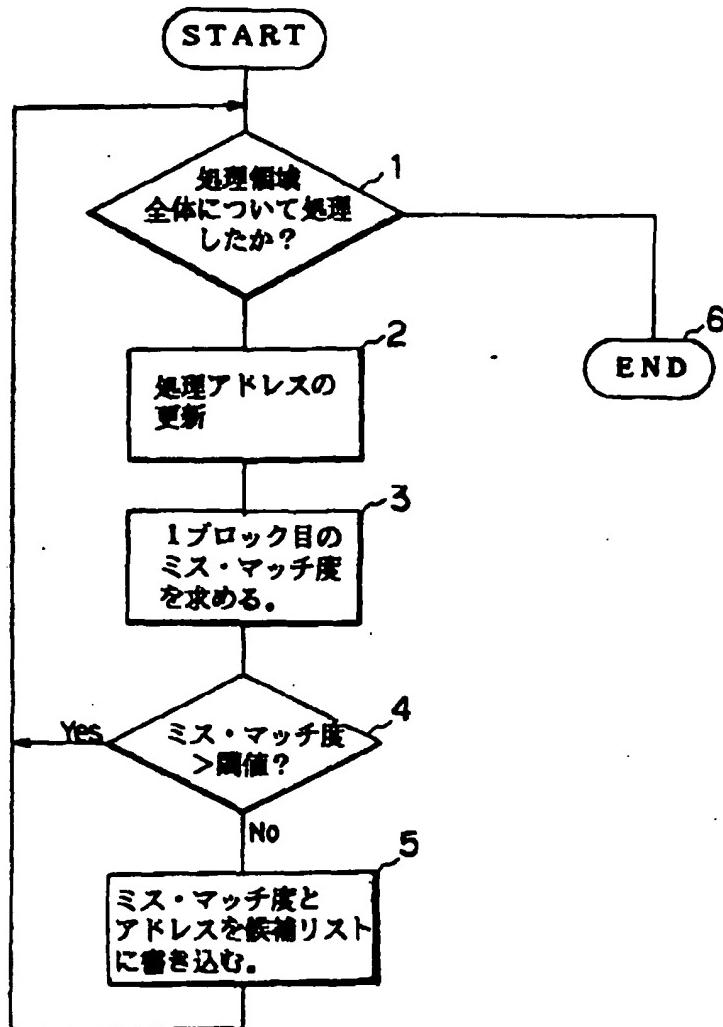
【図4】



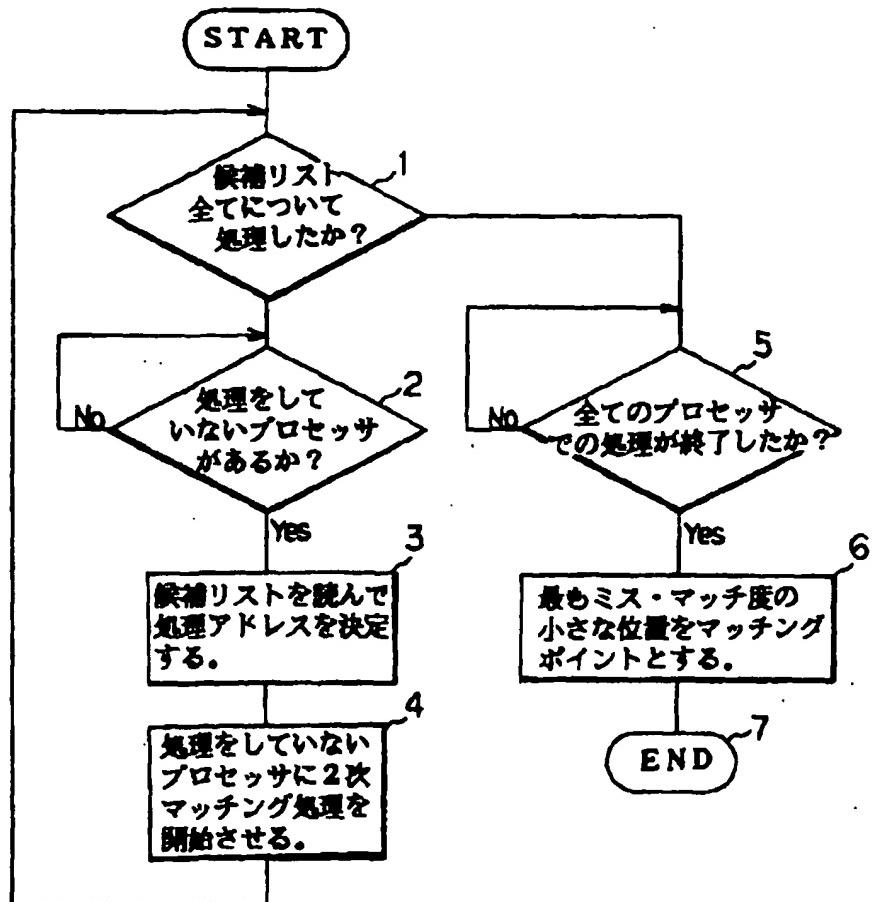
【図7】



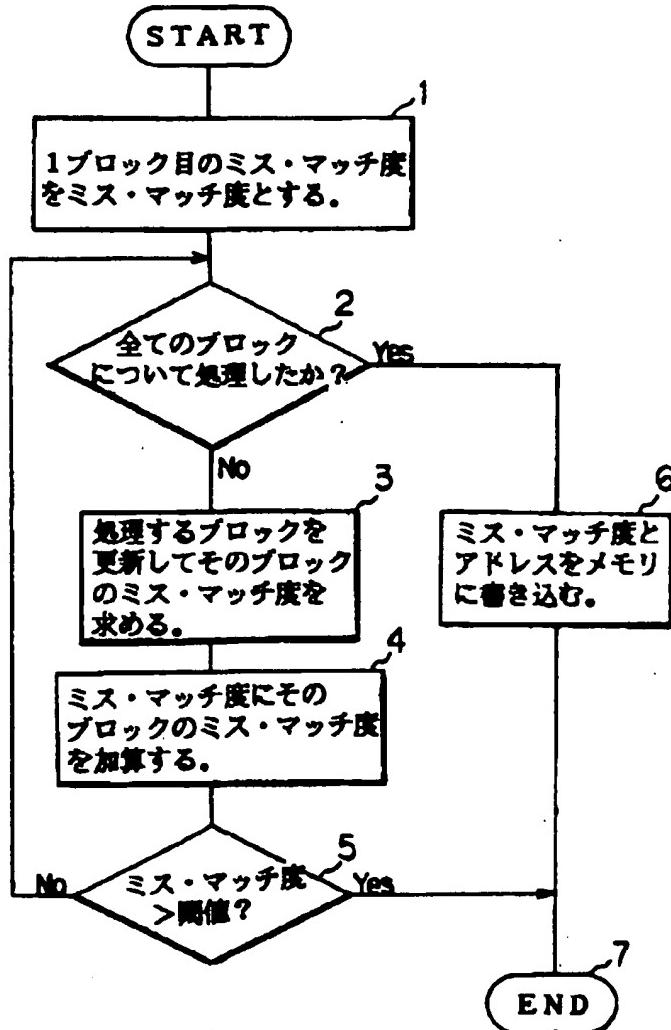
〔図6〕



[図10]



【図11】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER: _____**

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.